Aplicação prática App Inventor

APRESENTAÇÃO

Você já observou em seu dispositivo, seja ele *smartphone* ou *tablet*, a quantidade de aplicativos que são utilizados? São muitos, e com diversas funcionalidades, já que atualmente, atrelam-se as atividades cotidianas ao uso desses recursos, de modo a ter praticidade no dia a dia. Você já se questionou em como esses aplicativos são desenvolvidos? Geralmente, além de ferramentas específicas, o seu desenvolvimento depende de equipes multidisciplinares, ou seja, não são compostas apenas por programadores, mas por *designers*, gestores de projeto, analistas, etc.

Nesta Unidade de Aprendizagem, você aprenderá a desenvolver aplicativos por meio da ferramenta *web* de desenvolvimento App Inventor 2, a qual é atualmente mantida pelo MIT, tendo o objetivo de tornar o desenvolvimento de *software* algo lúdico e atrativo para qualquer tipo de desenvolvedor, seja ele experiente ou iniciante.

Bons estudos.

Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Produzir um aplicativo.
- Empregar os diferentes recursos oferecidos pelo App Inventor.
- Usar o App Inventor Companion para testar um aplicativo.

INFOGRÁFICO

As ferramentas utilizadas para desenvolvimento de *software* trazem diversos benefícios para o desenvolvedor, já que este terá que lidar com a complexa função de desenvolvê-lo. Entre esses benefícios, podem ser citados os recursos, os quais, no caso do App Inventor, são os componentes da Paleta, que auxiliam na concepção do *layout* do aplicativo.

Neste Infográfico, você vai ver todos os componentes da Paleta do App Inventor.

CONHECENDO OS RECURSOS DA PALETA

A Paleta traz recursos para o desenvolvimento do layout do aplicativo, os quais são separados por áreas.

INTERFACE DO USUÁRIO

A interface do usuário traz componentes voltados para as ações que o usuário pode desempenhar no aplicativo, como botões, caixa de texto, imagens, caixas para inserção de senhas, etc.

ORGANIZAÇÃO

Os componentes de Organização também são chamados de componentes de *layout*. Estes proporcionam ao desenvolvedor a escolha do posicionamento dos elementos no *layout* das telas do aplicativo.

MÍDIA

Os componentes de Mídia permitem que o aplicativo interaja com o usuário por meio do uso da câmera, câmera de vídeo, som, gravador, entre outros recursos. Eles deixam o aplicativo mais interativo.

DESENHO E ANIMAÇÃO

Os componentes de Desenho e Animação permitem, por exemplo, que você insira alguma imagem de fundo para o seu aplicativo. Estes possibilitam a interação do usuário com o aplicativo e entre os próprios componentes.

MAPAS

Os componentes de Mapas possibilitam o uso de mapas e informações geográficas conforme a necessidade do desenvolvedor.

SENSORES

Os componentes de Sensores possibilitam, por exemplo, a detecção de quando o dispositivo está em movimento.

SOCIAL

Esses componentes permitem que o aplicativo forneça funcionalidades de interação social, como ligação, envio de *e-mail*, compartilhamento, etc.

Social
EscolheContato
EscolheEmail
Ligação
EscolheNúmeroDeTelefone
Compartilhamento
MensagensSMS
Twitter

ARMAZENAMENTO

Esses componentes permitem o uso de armazenamento de informações, tanto por meio de um arquivo quanto por banco de dados.

Armazenamento
Arquivo
ControleDeFusiontables
TinyDB
TinyWebDB

CONECTIVIDADE

Esses componentes permitem que o aplicativo se conecte por meio de bluethooth, conexão do aplicativo com outros aplicativos e outros ambientes, como o web, por exemplo.

LEGO® MINDSTORMS NXT.

Esses componentes fornecem controle de robôs LEGO® MINDSTORMS® NXT usando *bluetooth*, por meio do aplicativo desenvolvido no App Inventor.

EXPERIMENTAL

Esses componentes permitem o armazenamento de dados. Como, por exemplo, o cloudDB permite armazenar dados em um servidor de banco de dados conectado à Internet (usando o *software* Redis). Isso permite que os usuários do seu aplicativo compartilhem dados entre si. Por padrão, os dados serão armazenados em um servidor mantido pelo MIT, mas você pode configurar e executar o seu próprio servidor.

EXTENSION

Esse componente permite importar extensões externas para que sejam utilizadas no aplicativo.

CONTEÚDO DO LIVRO

O App Inventor, assim como as demais ferramentas de desenvolvimento de *software*, traz muitos elementos disponibilizados para facilitar todo o processo, os quais consistem tanto em componentes voltados para o *layout* do aplicativo quanto para a aplicação da lógica de programação.

Para saber mais, acompanhe a leitura do capítulo Aplicação prática App Inventor, da obra *Pensamento computacional*, base teórica desta Unidade de Aprendizagem.

Boa leitura.

PENSAMENTO COMPUTACIONAL

Izabelly Soares
de Morais

Aplicação prática App Inventor

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Produzir um aplicativo.
- Empregar os diferentes recursos oferecidos pelo App Inventor.
- Usar o App Inventor Companion para testar o aplicativo.

Introdução

Apesar de termos diversos aplicativos disponíveis nas lojas virtuais dos nossos sistemas operacionais — Google Play Store, App Store (iTunes), dentre outras —, temos uma grande chance de não encontrar um aplicativo que sane toda a nossa necessidade. Observe que os aplicativos visam a solucionar problemas comuns a todas as pessoas. Em contrapartida, o que te impede de desenvolver o seu próprio aplicativo?

Neste capítulo, vamos abordar aspectos fundamentais para o desenvolvimento de aplicativos utilizando a ferramenta App Inventor, que se encontra em sua segunda versão. Essa é uma ferramenta lúdica que proporciona ao desenvolvedor uma melhor compreensão de toda a lógica de programação aplicada durante o processo de desenvolvimento. Essa compreensão é realizada por meio de bloco de instruções.

Etapas para produção de um aplicativo

O desenvolvimento de um aplicativo demanda dedicação, pois ter a ideia não é suficiente. Atualmente, as equipes de desenvolvimento são multidisciplinares, ou seja, não são compostas apenas por programadores, e sim por *designers*, gestores de projeto, analistas, dentre outros.

Outro fator bastante importante, além da formação da equipe, é o próprio usuário, ou seja, o público-alvo de seu aplicativo. Já que, ao criar um aplicativo,

estamos desenvolvendo algo que será utilizado com alguma finalidade, então, é recomendável que possamos proporcionar uma boa experiência ao usuário. Um dos pontos cruciais para que isso ocorra é a organização do *layout*. Mas o que é *layout*? Podemos compreender como *layout* a organização dos componentes do aplicativo, como botões, textos, imagens, vídeos, dentre outros. Ele proporciona uma harmonização de como tudo isso será exposto ao usuário.

Essa ideia é elencada não só para aplicativos, mas para todo tipo de desenvolvimento. Dessa forma, devemos ficar atentos a algumas observações, as quais foram levantadas por Kalbach (2009), mas que iremos adaptá-las ao nosso contexto, que é o de desenvolvimento de aplicativos. Existem três áreas principais de interesse no processo de *layout*, os quais veremos a seguir.

Determinar os caminhos de navegação

Ao selecionar os menus e os mecanismos para sua navegação, identifique as rotas ótimas nas quais as pessoas provavelmente viajarão para alcançar seu conteúdo-chave e modele a navegação em torno dessas rotas. Utilize personas e cenários para ajudar na escolha dos mecanismos, de forma a evitar projetar demasiadamente o sistema de navegação.

Fique atento

Personas são descrições narrativas de arquétipos de usuário refletindo padrões de necessidades e comportamentos descobertos durante a inteligência do usuário. O cenário é outra maneira eficiente e efetiva de refletir uma grande quantidade de informações descobertas a partir de atividades de inteligência do usuário, eles são descrições detalhadas do que o *site* deveria fazer a partir da perspectiva do usuário. Eles são importantes para descrever a experiência do usuário. Os cenários podem comunicar rapidamente para uma equipe de desenvolvimento ou para *stakeholders* a sua visão de como o site será usado (KALBACH, 2009).

Projetar uma lógica visual

Uma vez que souber a quantidade de navegação necessária e os diferentes mecanismos envolvidos, é possível então começar a organizar estes nas pá-

ginas. O objetivo é criar um padrão reconhecível de elementos que facilitem a navegação no aplicativo.

Templates de páginas

Um sistema de *templates* de navegação é importante para a consistência do *design*, bem como para a eficiência da implementação. *Templates* de página capturam os seus *layouts* e mostram uma progressão de mudanças de uma página/tela a outra.

Utilizaremos aqui o App Inventor, pois é uma ferramenta *web* que proporciona um ambiente de desenvolvimento lúdico, o que a diferencia das demais ferramentas de desenvolvimento. O lúdico se aplica na possibilidade de podermos desenvolver um aplicativo utilizando blocos de instruções, além disso, o *layout* também já traz componentes prontos, em que o desenvolvedor precisa apenas selecionar o que quer e arrastar para a área do visualizador, como mostra a Figura 1 a seguir.

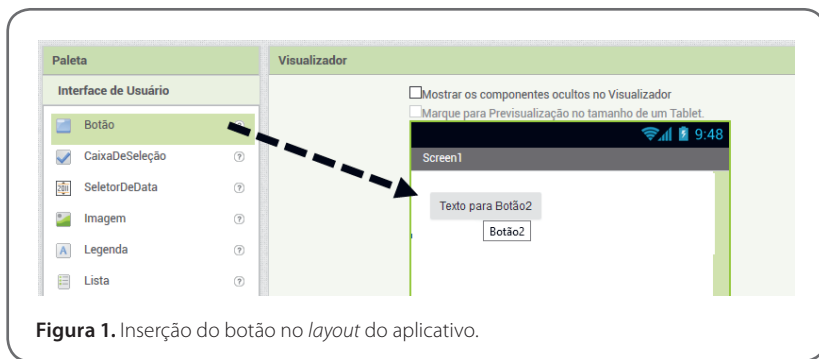



Figura 1. Inserção do botão no *layout* do aplicativo.

A utilização da ferramenta é bem simples, por ser *web*, você precisa inicialmente acessar o *site* da ferramenta. Observe que no canto superior direito tem a opção *Crie aplicativos!*, que estará na língua inglesa, já que a ferramenta é mantida pelo Instituto de Tecnologia de Massachusetts (MIT), então, provavelmente, estará como *Create apps!*. Dependendo do navegador que você estiver utilizando, você pode colocar para traduzir automaticamente a página inicial, já que, quando você realizar o *login*, poderá escolher o idioma que deseja na própria ferramenta. Veja a Figura 2.

Link

Acesse o site do App Inventor para utilizar a ferramenta:

https://goo.gl/2bnlh

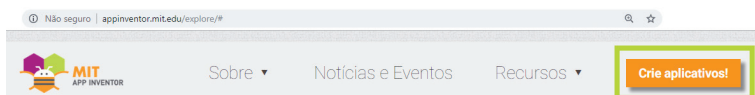



Figura 2. Fazendo o *login* no App Inventor.

Falamos em *layout* anteriormente e é preciso pensarmos um pouco nele, antes de começarmos o desenvolvimento do nosso aplicativo. Devemos pensar também em quais funcionalidades queremos que ele tenha, para que possamos planejar tudo com calma. Agora que já comentamos um pouco sobre a ferramenta, vamos aprender na prática sobre alguns recursos que a ferramenta disponibiliza para que possamos desenvolver nossos aplicativos.

Recursos para criação de aplicativo

O App Inventor tem dois ambientes de desenvolvimento, um chamamos de *Designer* e o outro, de *Blocos*. À medida que formos utilizando os ambientes, vamos conhecendo cada vez mais suas particularidades. Agora, vamos focar no *Designer*. É neste ambiente que o desenvolvedor insere os componentes que farão parte do *layout* do aplicativo, ou seja, quais funcionalidades suas telas terão: botões, escolha das cores, textos, etc. Há muitos detalhes para pensar em relação ao *layout*. Um ponto positivo é que a ferramenta auxilia nessas decisões, pois ela disponibiliza um campo com todos os componentes disponíveis, chamado de *Paleta* (ou *Palette*, em inglês). Na Figura 3, observa-se a *Paleta*.

Figura 3. Paleta e seus componentes.

Podemos notar que a Paleta se subdivide em outras seções. Nela podemos visualizar componentes que vão desde a interface do usuário ao uso de extensões externas a ferramenta. O objetivo disso tudo é justamente nos proporcionar um ambiente o mais completo possível. Temos diversos recursos, os quais podemos ou não fazer uso, e isso dependerá da nossa necessidade. Todas as abas da Paleta são extensíveis. Ao clicarmos em *Interface do usuário*, veremos os componentes relacionados a esse contexto, como podemos visualizar na Figura 4 a seguir.

Link

Mais informações sobre os componentes e suas respectivas particularidades podem ser acessadas por meio do link a seguir.

https://goo.gl/YMKzdj

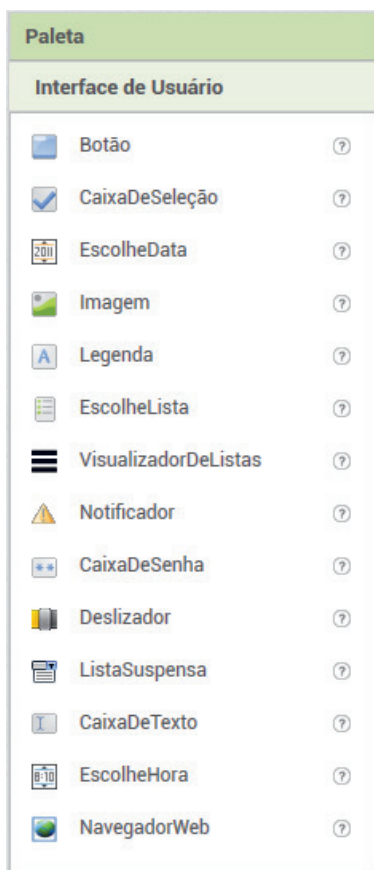



Figura 4. Paleta — componentes da interface do usuário.

Aqui, podemos escolher qual componente queremos e arrastar para o visualizador. Veja que temos botões, caixa de seleção, campo para escolha de data, dentre outras funcionalidades.

É importante você saber que a ferramenta também traz um campo em que as propriedades dos componentes podem ser modificadas, como mostra a Figura 5 a seguir.

Figura 5. Propriedade do componente *Botão*.

Estamos utilizando como exemplo as propriedades do componente *Botão*, mas saiba que todos os componentes têm propriedades e que elas podem ser alteradas conforme a necessidade demonstrada ao desenvolvedor no momento em que o aplicativo está sendo desenvolvido. Observe que podemos alterar cor, tamanho da fonte, largura, dentre outros. Claro que as propriedades variam de acordo com o componente. Não esqueça de que para ficarem visíveis você deve selecionar o componente antes, pois, com certeza, trabalharemos com muitos ao mesmo tempo, então precisamos definir em qual gostaríamos de modificar algo.

Abaixo dos componentes da interface do usuário, temos os de *layout*, ou de *organização* (esse termo é visto quando escolhemos a língua portuguesa para a ferramenta). Veja a Figura 6.

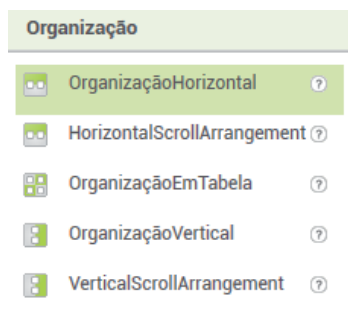



Figura 6. Componentes de *layout*.

Esses componentes permitem que haja organização no *layout* geral do aplicativo, por exemplo, a organização horizontal vai mostrar os componentes da esquerda para a direita, enquanto a organização vertical mostra os componentes da direita para a esquerda. Observe que os próprios ícones trazem uma demonstração de suas funcionalidades. Seguindo a ordem dos componentes da Paleta, temos os componentes de mídia. Veja a Figura 7.

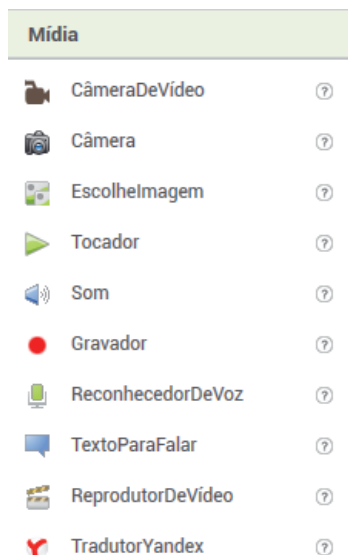



Figura 7. Componentes de mídia.

Atualmente esses componentes são bem importantes e enriquecem o aplicativo, tendo em vista que hoje o que o usuário mais deseja é interagir. Esses componentes de mídia proporcionam justamente isso, já que traz o uso de câmera, câmera de vídeo, uso de imagens, som, gravador, dentre outros recursos.

Ainda na Paleta, temos acesso aos componentes de *Desenho e animação*, como podemos ver na Figura 8 a seguir.

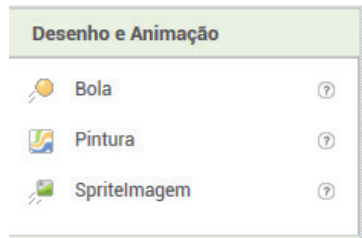



Figura 8. Componentes de desenho e animação.

Esses componentes são utilizados para gerar interação entre o usuário, componentes e aplicativo. Por exemplo, você pode inserir uma imagem de fundo usando o componente *Pintura*, lembrando que as configurações de cada componentes serão realizadas em suas propriedades. Os próximos componentes são os de mapas. Veja a Figura 9.

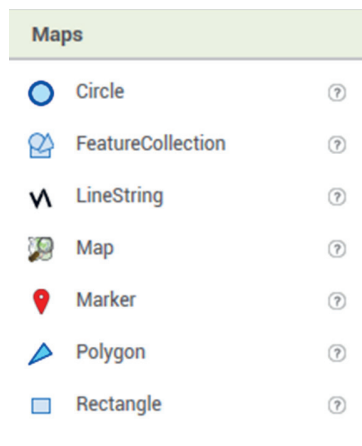



Figura 9. Componentes de mapas.

Apesar de a seção ter o nome *Mapas*, o desenvolvedor pode inserir outras funcionalidades ligadas a informações geográficas, além de, claro, poder inserir um mapa em seu aplicativo. Observe a Figura 10.

Figura 10. Componentes de sensores.

Os sensores permitem que o aplicativo também interaja com o usuário e, além disso, que interaja com o meio, já que temos como opção sensor de localização, de orientação e de proximidade e até mesmo leitor de código de barras. São muitas funcionalidades, só depende de você, desenvolvedor. O que desejará utilizar? Será que se acrescentar mais alguma funcionalidade ao seu aplicativo, ele não ficará mais interessante?

Não esqueça que tudo deve ser coerente, também não adianta você inserir muitas funcionalidades desconexas umas com as outras. Veja a Figura 11.

Figura 11. Componentes de social.

O App Inventor se renova a cada versão, muitas vezes com funcionalidades novas, ou com possibilidades de utilizarmos as já existentes de outra forma. Nos componentes sociais temos ligações, compartilhamento, escolha de e-mail, dentre outros. São formas que o usuário do aplicativo terá de socializar com outras pessoas por meio desses recursos. Veja a Figura 12.

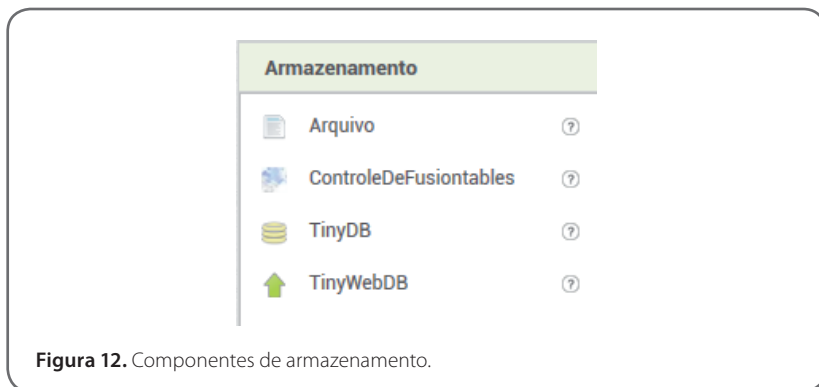
Figura 12. Componentes de armazenamento.

Os aplicativos estão cada vez mais robustos, ou seja, trazendo mais funcionalidades e consequentemente exigindo mais recursos de suporte, por isso os componentes de armazenamento são tão importantes. Eles possibilitam que haja armazenamento e recuperação de arquivos, leitura de arquivos no aplicativo, dentre outros. Observe a Figura 13.

Figura 13. Componentes de conectividade.

Esses componentes permitem que o aplicativo se conecte por meio de *Bluetooth*, conexão do aplicativo com outros aplicativos e outros ambientes, como *web*, por exemplo. Têm também componentes invisíveis para prover funções para as requisições de baixo nível HTTP do tipo GET, POST, PUT e DELETE, dentre outros. Veja a Figura 14.

LEGO® MINDSTORMS®

NxtAcionar
NxtSensorDeCor
NxtSensorDeLuz
NxtSensorDeSom
NxtSensorDeToque
NxtSensorUltrasônico
NxtComandosDiretos
Ev3Motors
Ev3ColorSensor
Ev3GyroSensor
Ev3TouchSensor
Ev3UltrasonicSensor
Ev3Sound
Ev3UI
Ev3Commands

Figura 14. Componentes para uso com LEGO.

Um outro componente é o LEGO® Mindstorms NXT. Esses componentes fornecem controle de robôs LEGO® Mindstorms NXT usando *Bluetooth*, por meio do aplicativo desenvolvido no App Inventor. A partir dessa conexão, os componentes podem prover interface alto nível para o sensor de som no robô LEGO Mindstorms NXT, para sensor de luz no robô LEGO Mindstorms NXT, dentre outros. Veja a Figura 15.

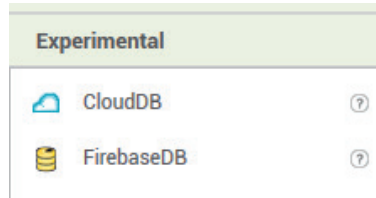



Figura 15. Componentes experimentais.

Anteriormente, vimos os componentes experimentais, eles permitem o armazenamento de dados. Por exemplo, o *CloudDB* permite armazenar dados em um servidor de banco de dados conectado à internet (usando o *software* Redis). Isso permite que os usuários do seu aplicativo compartilhem dados entre si. Por padrão, os dados serão armazenados em um servidor mantido pelo MIT, mas você pode configurar e executar seu próprio servidor. Veja a Figura 16.

Figura 16. Componentes para uso de extensões.

Por fim, temos a possibilidade de importar alguma extensão externa para utilizar no aplicativo. Adicione como funcionalidade *Bluetooth Low Energy* aos seus aplicativos e adicione reconhecimento de objetos usando uma rede neural compilada na extensão.

Agora que exploramos um pouco os componentes da Paleta, vamos praticar. Só assim conseguiremos, aos poucos, aprender cada vez mais sobre a ferramenta e como ela funciona.

Testando o aplicativo com o App Inventor Companion

A princípio, vamos criar um aplicativo básico, para que possamos nos familiarizar com a ferramenta e com o App Inventor Companion.

Link

Para testar o App Inventor Companion, acesse o link a seguir.

http://ai2.appinventor.mit.edu/

Acesse o *site* e selecione a opção de *Criar aplicativo*, como mencionamos anteriormente. Nesse exemplo prático, vamos criar uma calculadora básica, que realizará a operação de adição.

Após a criação de seu novo projeto, o primeiro passo será a inserção de um plano de fundo, para que nosso aplicativo tenha uma aparência mais atrativa para o usuário. Observe que já temos uma primeira tela, a ferramenta *Screen1*. Nessa tela, iremos inserir a imagem de fundo. Para isso, observe que ao lado direito, em *Propriedades*, temos a opção *ImagemDeFundo*, como mostra a Figura 17 a seguir. Não esqueça de selecionar o componente na barra *Componentes*.

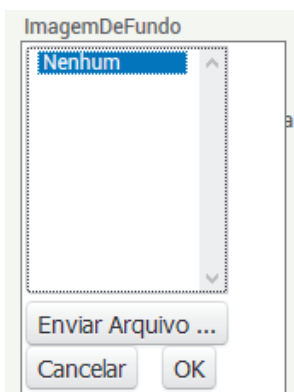



Figura 17. Propriedades da *Screen1*.

Uma observação é que a ferramenta só aceita arquivos que não contenham números ou caracteres alfanuméricos em seus nomes. Uma outra dica é a do uso de imagens no formato **.png**, pois têm uma qualidade melhor na compressão de arquivos. Dessa forma, escolha a imagem que achar melhor. Para isso é só clicar em *Enviar arquivo*, selecionar a imagem e finalizar. Por fim, na aba de visualizador, você verá que o *layout* de seu aplicativo mudou e a imagem de fundo que você escolheu está visível. Veja a Figura 18.

Figura 18. *Layout* da tela do aplicativo após inserção de imagem de fundo.

Outro passo que podemos dar é mudar o nome da tela, o qual fica localizado na barra de título, em vez de *Screen1*, podemos colocar o nome que acharmos melhor; uma sugestão é o nome *Calculadora*. Essa mudança pode ser realizada em propriedades e depois no campo *Título*, como mostra a Figura 19 a seguir.

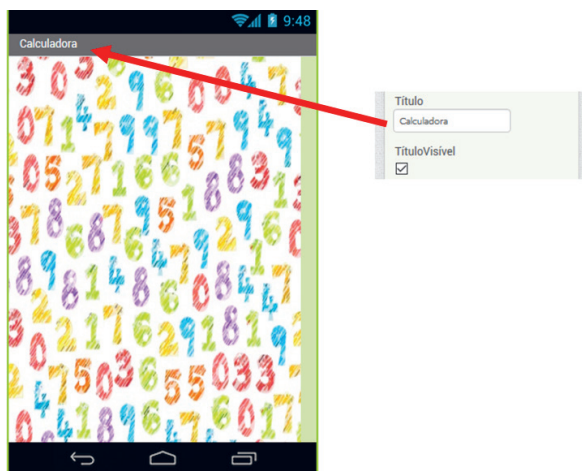



Figura 19. Campo para mudança de título da tela.

Inclusive, após o espaço para mudança do título, temos a opção de deixa-lo ou não visível no *layout* do aplicativo. Esses detalhes ficam a seu critério.

Citamos anteriormente que os componentes do *layout*, ou de organização, nos auxiliarão na organização visual dos nossos componentes. Dessa forma, vamos fazer uso desses componentes nesse momento, já que vamos começar a inseri-los na nossa tela inicial. Veja a Figura 20.

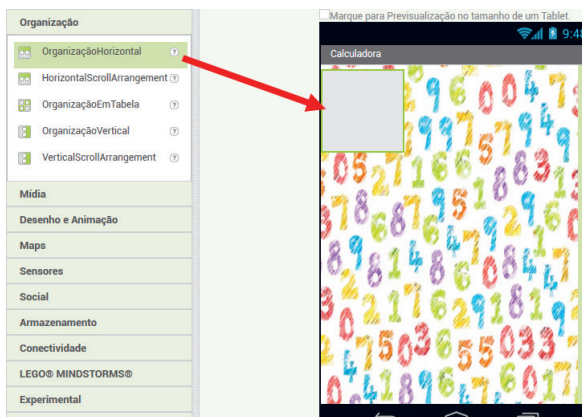



Figura 20. Uso do componente de organização.

Vamos selecionar a *OrganizaçãoHorizontal*, que é a primeira opção, que possibilita que os elementos sejam organizados de forma horizontal e permite também que sejam postos uns ao lado dos outros. Arraste o componente e o posicione no *layout* de seu aplicativo, localizado no *Visualizador*. Agora vamos realizar algumas configurações, conforme a Figura 21 a seguir.

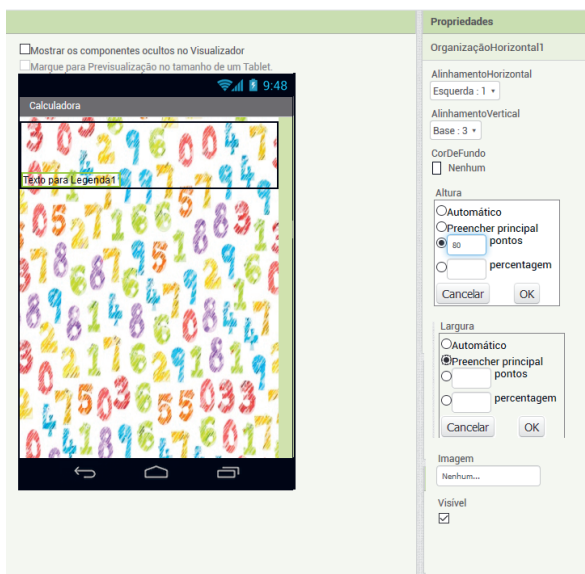



Figura 21. Configurações no *layout*.

Selecione na propriedade *Screen1* o *AlinhamentoVertical*, no qual você deverá escolher a *Base:3*. Essa configuração foi realizada para permitir que possamos inserir componentes na posição vertical. Além disso, alteramos também a altura para 80, que, na tradução, fica com o termo **pontos**. Porém, esse campo remete aos *pixels*, ou seja, a proporção que o campo irá tomar para testá-lo. Experimente inserir outros valores e veja o que ocorre. Na largura, selecionamos a opção de *Preencher principal*, na prática, ele preenche todo o campo na posição horizontal, como mostra a Figura 21 anterior. Veja que o quadro verde está tomando toda a proporção da tela. Sobre a cor de fundo, deixamos ***nenhuma***, para que a imagem de fundo que colocamos fique aparente.

Saiba mais

A propriedade *AlinhamentoHorizontal* é utilizada para alinhar os componentes em seu interior e temos a opção de deixá-los à esquerda (para isso, escolhemos a opção *Esquerda:1*), centralizados (usamos a opção *Center:3*) e à direita (utilizamos a opção *Direita:2*). A propriedade *AlinhamentoVertical* alinha os componentes de maneira vertical e temos a opção de deixá-los no topo (para isso escolhemos a opção *Topo:1*), no centro (opção *Centro:2*) e na base (opção *Base:3*).

Na Paleta, em *Interface do usuário,* selecione o componente *Legenda* e insira dentro do componente de *OrganizaçãoHorizontal*, que acabamos de configurar. A seguir, na Figura 22, estão algumas configurações aplicadas ao componente *Legenda* (*Label*, em inglês).

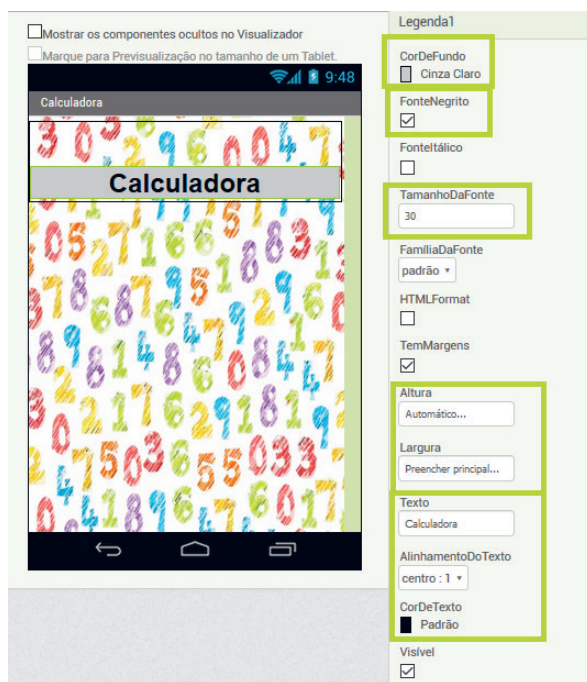



Figura 22. Configurações nas propriedades do componente *Legenda*.

Na Figura 22 anterior, os campos das propriedades do componente *Legenda* foram selecionados, mas é importante ressaltar, novamente: crie as configurações que julgar necessárias. Como estamos usando uma imagem de fundo estampada, há as seguintes alterações: a cor do fundo foi mudada para cinza; selecionamos a opção de deixar o texto em negrito, para que ele fique mais destacado; mudamos também o tamanho da fonte para 30; para a altura selecionamos a opção automática; e para a largura selecionamos a opção de mantê-la preenchida, ou seja, *Preencher principal*. Observe que você poderá mudar o texto e escolhemos inserir o texto *Calculadora*. Abaixo, você também seleciona o alinhamento do texto e escolhe *Centro:1* para que o texto possa ficar centralizado. Por fim, mudamos a cor do texto para preto.

Os próximos passos serão criar os campos para que o usuário possa inserir os valores e, claro, teremos que utilizar outro *OrganizaçãoHorizontal*, que se encontra em *Paleta*, na seção *Organização*. Veja a Figura 23.

Figura 23. Alterando a propriedade de outro componente *OrganizaçãoHorizontal.*

As configurações foram as mesmas utilizadas anteriormente, no componente *OrganizaçãoHorizontal*. Para o *AlinhamentoHorizontal*, escolhemos a opção *Centro:3* (indica que todos os componentes que estarão dentro de *OrganizaçãoHorizontal* serão alinhados ao centro), e no *AlinhamentoVertical*, escolhemos a opção *Centro:2* (indica que todos os componentes estarão dentro

de *OrganizaçãoHorizontal* serão alinhados ao centro). Após alinharmos os elementos, escolhemos retirar a cor de fundo. Para isso, escolha a opção ***nenhum***. Por fim, alteramos a largura para *preencher principal*, ou seja, para que o componente ocupe toda dimensão horizontal do aplicativo, como mostra a Figura 23 anterior.

Vamos inserir no componente de *OrganizaçãoHorizontal* que acabamos de configurar um componente do tipo *Legenda* e outro do tipo *CaixaDeTexto*. Eles se encontram na *Paleta*, em *Interface do usuário*. Veja a Figura 24.

Figura 24. Uso dos componentes *Legenda* e *CaixaDeTexto*.

Agora que inserimos os elementos, vamos configurar suas propriedades.

Inicialmente, vamos selecionar, no Visualizador, o elemento que iremos configurar, no caso, começaremos pela legenda. Primeiro, vamos selecionar a opção *FonteNegrito*, para que o texto fique com destaque. Para isso, vamos aumentar o tamanho da fonte para 30. O próximo passo é alterarmos o texto da legenda. O texto foi alterado para *Número*, já que este será o espaço destinado para o usuário digitar um dos valores para realizar a execução da operação matemática. Alteramos também a cor de fundo para a cor rosa, como mostra a Figura 25 a seguir.

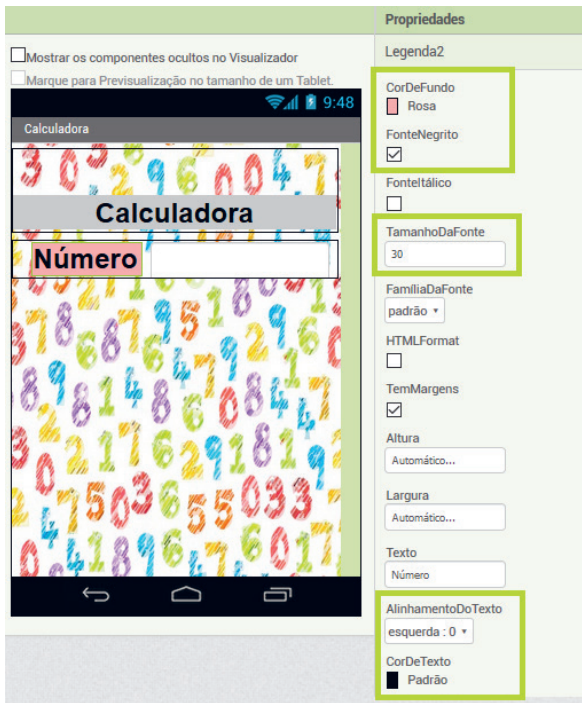



Figura 25. Propriedades do componente *Legenda*.

Fique atento

Não esqueça de selecionar o componente. Só assim as propriedades ficarão visíveis para alterações. Caso deseje renomear o nome dos componentes, você precisa apenas selecioná-lo em *Componentes* e clicar em *Renomear* (Figura 26).

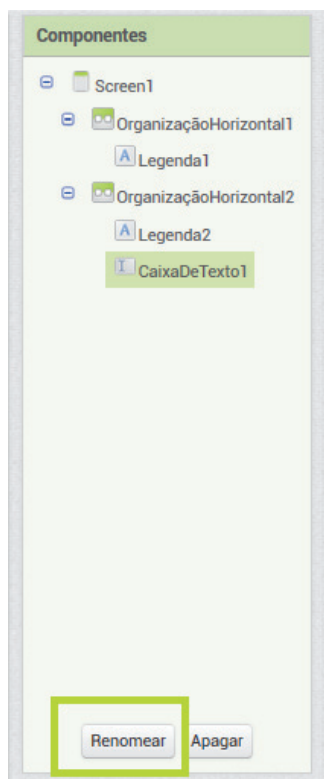



Figura 26. Opção de renomear os componentes.

Vamos agora para o componente *CaixaDeTexto*, o qual tem uma propriedade chamada *Dica*, na qual você pode inserir algo referente à funcionalidade que a caixa de texto terá, e ela ficará visível no momento em que o cursor for posicionado em cima da caixa. Aqui, foi inserida a seguinte dica: *Espaço para inserir um valor*, porém, observe que ao lado da caixa de texto já colocamos um texto com o objetivo da caixa, que é receber um número. Como nosso objetivo é que esse espaço receba apenas números, devemos selecionar a opção *SomenteNúmeros* em suas propriedades. Para entender, veja a Figura 27.

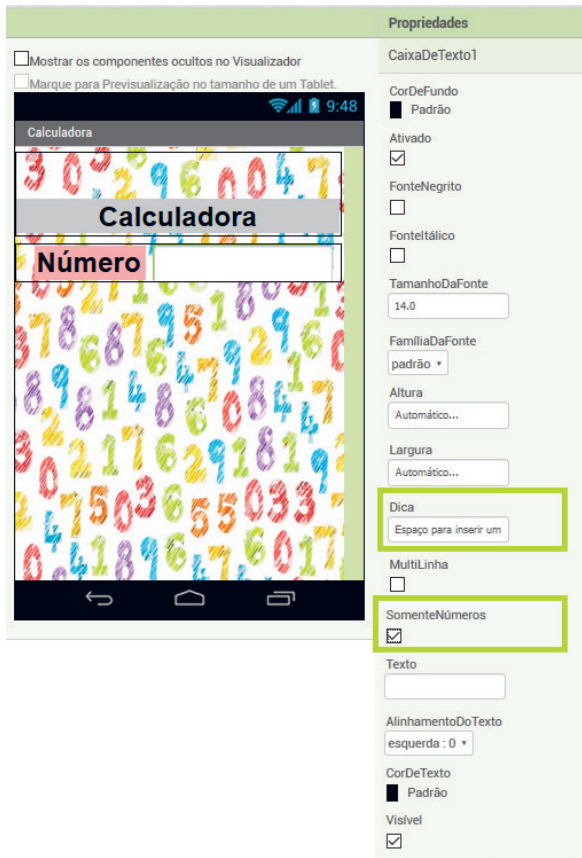



Figura 27. Propriedades do componente *CaixaDeTexto*.

Saiba mais

A propriedade da *CaixaDeTexto*, denominada como *SomenteNúmeros*, aciona um teclado numérico para que o usuário tenha a opção de digitar apenas números. A digitação de caracteres, como letras, por exemplo, causaria erro de execução do aplicativo.

Já que iremos realizar operações matemáticas com dois valores, repita a inserção do campo de texto e da legenda que fizemos para o primeiro valor, sendo que agora você criará outros componentes dos mesmos tipos e configurações para receber o segundo valor. O *layout* deverá ficar igual ou semelhante ao mostrado na Figura 28 a seguir.

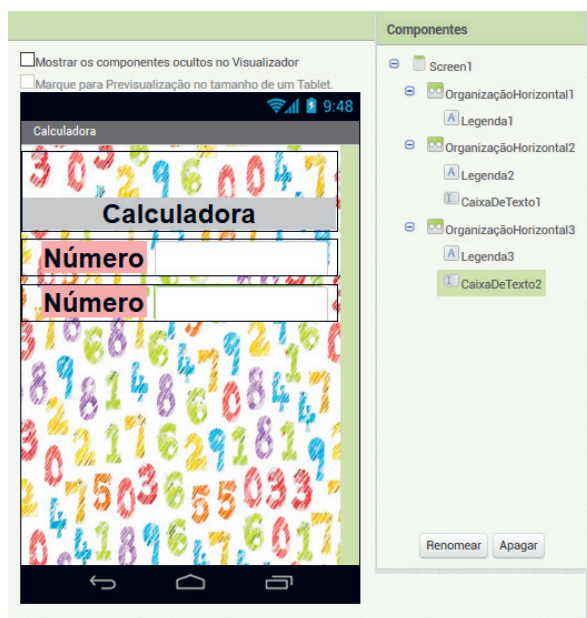



Figura 28. *Layout* do aplicativo.

Seguindo nosso objetivo, o próximo passo será inserir as opções referentes às operações matemáticas que nosso aplicativo estará apto a realizar. Para organizar melhor o *layout*, vamos inserir outro componente *OrganizaçãoHorizontal*, para que tenhamos espaço para inserir outros componentes. O objetivo desse componente será apenas o de dar espaçamento, por isso, devemos alterar suas propriedades. Primeiro vamos deixa-lo sem cor de fundo, selecione em *CorDeFundo* a opção *nenhum*. Na propriedade *Altura*, insira o valor 40 pontos (ou seja, *pixels*), e na propriedade *Largura*, altere para *preencher principal*.

Até agora inserimos espaços para que o usuário insira seus valores, inserimos componente para organizar melhor o *layout*, e agora daremos continuidade ao nosso aplicativo. Para isso, vamos novamente na Paleta inserir componente de organização. Utilizaremos o mesmo que estamos utilizando até o momento,

que é o de *OrganizaçãoHorizontal*. As configurações devem ser: na propriedade *AlinhamentoHorizontal*, devemos escolher a opção *Centro:3*, e na propriedade *Largura*, devemos selecionar *preencher principal*.

Voltando à Paleta, na seção *Interface do usuário*, selecione o componente *Botão*, posteriormente arraste-o e o insira dentro do componente de *OrganizaçãoHorizontal*, que colocamos anteriormente no *layout* de nosso aplicativo. Esse botão terá como funcionalidade realizar as operações matemáticas que definirmos.

Após a inserção do componente, vamos alterar suas propriedades. Inicialmente, vamos alterar o tamanho de sua fonte, inserindo na propriedade *TamanhoDaFonte* o valor 30. Para harmonizar com os demais componentes, vamos inserir o valor de 50 pontos para a altura e 40 pontos para a largura. No texto, iremos inserir o símbolo da operação matemática, que, no caso, iniciaremos com a adição, então o símbolo é o de +. Em vez de inserir um texto no botão, você também poderá adicionar alguma imagem. A Figura 29 a seguir mostra como ficará o *layout* após essas configurações.

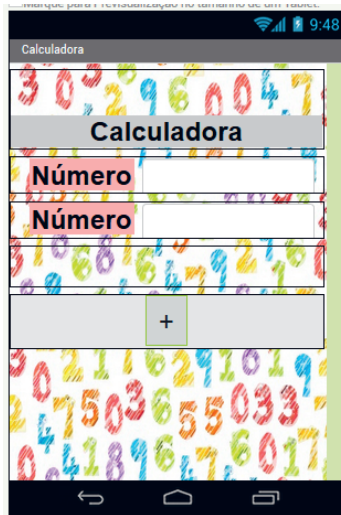



Figura 29. Inserção do botão para adição.

Na Figura 30 a seguir, podemos visualizar como o *layout* está ficando organizado.

Figura 30. *Layout* do aplicativo.

Na Figura 30 anterior, podemos observar que o *layout* do aplicativo está tomando forma, podemos notar também como é o teclado numérico que surge, ao selecionarmos a caixa de texto, em que o nosso objetivo será o de inserir apenas números.

A exibição do resultado será exposta em um componente do tipo legenda. Porém, antes, não esqueça que sempre que formos inserir um novo componente devemos pensar em sua organização diante dos demais componentes já existentes. Para isso, devemos inserir mais um componente do tipo *OrganizaçãoHorizontal*. Aqui você poderá escolher se quer inserir ou não esse componente, já que ele servirá apenas para organizar melhor o *layout*. Entretanto, você lembra que, anteriormente, inserimos um componente de organização entre o componente de mesmo tipo que receberá o segundo valor digitado e a operação matemática? Vamos aproveitá-lo, pois não alterará em nada nosso *layout*. Fique à vontade para criar ou não outro componente de organização, pois essa decisão é sua.

Como resolvermos reaproveitar o que já havíamos criado, inserimos agora dois componentes do tipo *Legenda*, um para receber um texto informativo de *Resultado* e o outro para mostrar para ao usuário o valor obtido como resultado. As configurações nas propriedades dos elementos foram apenas no texto e na cor de fundo. A Figura 31 a seguir traz o *layout* do aplicativo após a inserção desses elementos.

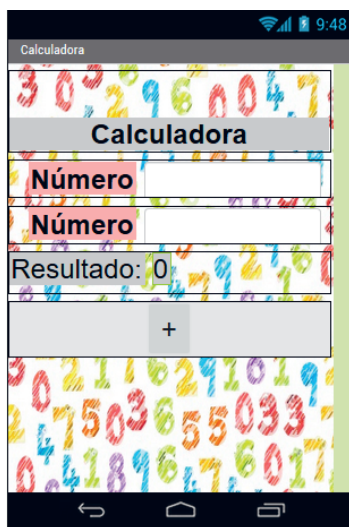



Figura 31. *Layout* do aplicativo.

As melhorias realizadas em relação ao *layout*, no sentido de configurações visuais, você pode realizar depois. O mais importante agora é inserirmos todos os elementos necessários para que nosso aplicativo possa realizar a soma de dois valores e nos apresentar o resultado. Agora, vamos para o ambiente de desenvolvimento de blocos.

Nesse ambiente, iremos configurar os espaços que criamos para receber os valores e, posteriormente, o botão que realizará o cálculo matemático. Para isso, vamos selecionar no menu *Blocos,* que fica localizado ao lado esquerdo da tela, o componente *Variáveis*.

Quando precisamos armazenar alguma informação no computador, realizamos essa operação por meio de uma variável. A ideia da variável, aqui, traz os mesmos conceitos da matemática, que é a de armazenar algum valor. Porém, quando criamos uma variável, um espaço é criado na memória do computador para receber esse valor, e para acessá-lo basta utilizar o nome que atribuímos à variável. Definimos dois espaços para receber dois números quando criamos dois componentes do tipo *CaixaDeTexto*, para isso, precisamos criar duas variáveis, uma para cada campo, como mostra a Figura 32 a seguir.

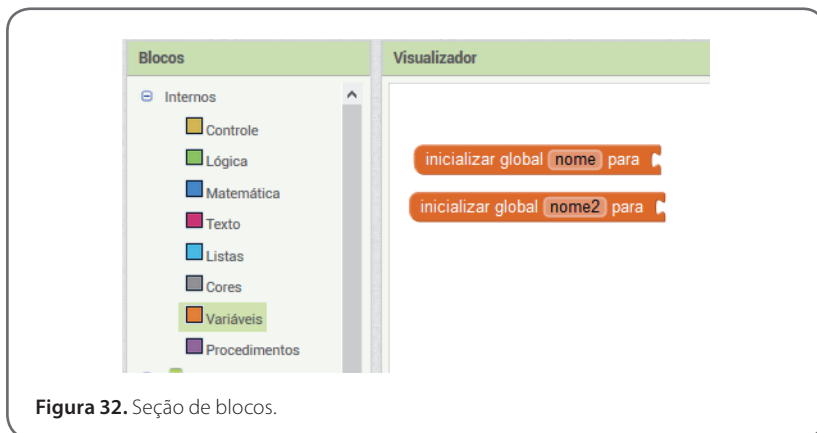



Figura 32. Seção de blocos.

Veja bem, iremos armazenar números, então o tipo das nossas variáveis será numérico. O tipo da variável remete ao tipo de dado que ela receberá. Você sabia que o tamanho criado na memória do computador para receber os valores das variáveis é definido conforme o tipo de dado que ela receberá? Por isso é importante sempre definirmos essa informação ao criarmos uma variável. Para inserir essa informação, vamos ao menu *Blocos*, selecionamos o componente de *Matemática* e selecionamos o primeiro bloco, que contém o 0 (zero). Vamos aproveitar e mudar o nome das variáveis. Para isso, basta clicar no espaço em que estão inseridos seus nomes e alterá-los. Veja a Figura 33.

inicializar global numero1 para 0

inicializar global numero2 para 0

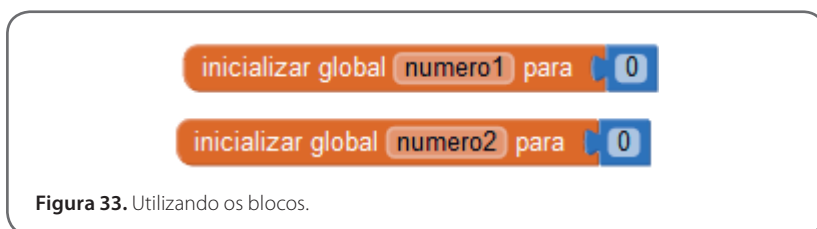



Figura 33. Utilizando os blocos.

O próximo passo será a definição das instruções que darão funcionalidade ao botão. Para isso, selecione no menu *Blocos* o componente que você criou como botão. Após a seleção, aparecerá no visualizador as opções de instruções que você pode atrelar ao botão. Como queremos que, após inserir os números e clicar no botão, ele execute a operação matemática, vamos selecionar a primeira opção, que é: *quando Botão1 .Clique fazer*. Essa opção executará algo assim que o botão foi clicado. Entenda melhor visualizando a Figura 34 a seguir.

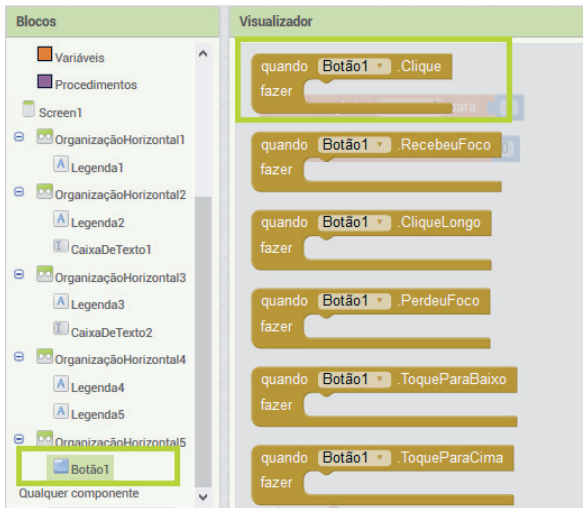



Figura 34. Utilizando os blocos.

Posteriormente, devemos voltar à seção *Variáveis* e selecionar o bloco *Ajustar para*. No espaço vazio, inserimos o nome da variável que armazenará o valor, como mostra a Figura 35 a seguir.

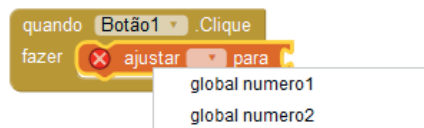



Figura 35. Configurando os blocos.

Devemos escolher a primeira opção. Essa variável global deverá receber o valor digitado no primeiro campo de texto que criamos. Para isso, vamos selecioná-lo no menu de blocos. Ao selecionarmos, aparecerá um menu de blocos de instruções, escolha a que terá o nome do seu campo de texto e depois *Texto*. Veja a Figura 36.

Figura 36. Configurando os blocos.

Para que o bloco de um componente *Legenda* possa exibir os valores, basta ajustar o nome do campo. A operação matemática deverá ser executada antes de exibirmos o valor. Para isso, vamos na seção de blocos de *Matemática* e selecionamos a opção de bloco que contenha espaço para executar tal ação. O encaixe após o bloco que contém o campo (que no nosso caso é o de nome *Legenda5*) exibirá o resultado do cálculo. Veja a Figura 37.

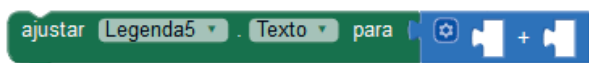



Figura 37. Blocos para a realização da soma.

Para selecionar as variáveis que irão preencher os espaços em branco, deve-se posicionar o *mouse* acima da variável global criada anteriormente, como mostra a Figura 38 a seguir.

Figura 38. Utilização dos blocos de instruções.

A opção *obter global numero1* indica que usaremos o valor que está armazenado na variável. Já a segunda opção, a de ***ajustar global numero1 para*** indica que iremos armazenar um novo valor na variável. Vamos selecionar a primeira opção e encaixá-la no espaço em branco. Deveremos atribuir essa mesma ação para as duas variáveis. Depois devemos encaixar esses blocos aos que criamos anteriormente. A configuração dos blocos ficará da seguinte maneira (Figura 39):

```
inicializar global numero1 para 0
inicializar global numero2 para 0

quando Botão1 .Clique
fazer  ajustar global numero1 para CaixaDeTexto1 . Texto
       ajustar global numero2 para CaixaDeTexto2 . Texto
       ajustar Legenda5 . Texto para obter global numero1 + obter global numero2
```

Figura 39. Utilização dos blocos de instruções.

O aplicativo ficará com o seguinte *layout* (Figura 40):

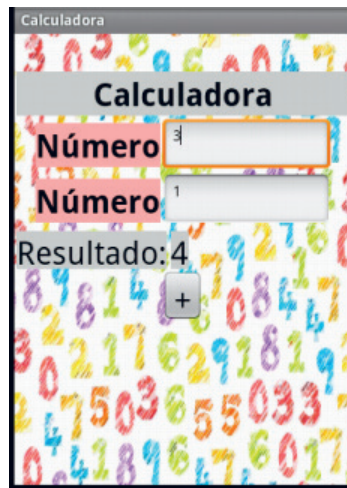



Figura 40. *Layout* final do aplicativo.

Alguns passos iniciais são primordiais para que você consiga realizar os testes de seu aplicativo no *Assistente AI*. Esse emulador pode ser seu próprio dispositivo, caso tenha o sistema operacional Android.

1. Acesse a Play Store em seu *smartphone* e digite "Mit app inventor 2".
2. Posteriormente, surgirão diversas opções de aplicativos relacionados ao termo procurado. Você irá selecionar a que tem este ícone (Figura 41):

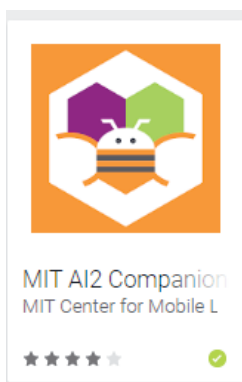



Figura 41. Ícone.

3. Enquanto você realiza o *download* em seu *smartphone*. Você deverá acessar na página *web* do App Inventor, no menu superior, a opção de conectar e posteriormente deverá selecionar a opção *Assistente Al*, como mostra a Figura 42 a seguir:

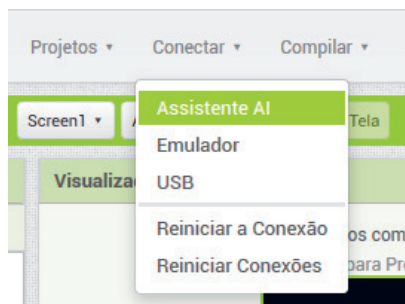



Figura 42. Seleção da opção *Conectar.*

Posteriormente, quando o *download* for finalizado e após você selecionar a opção *Assistente Al*, você será direcionado a uma página que trará um código de acesso, com seis dígitos e um QR Code. Você pode escolher se irá realizar a leitura do QR Code ou digitar o código. Caso escolha a leitura do QR Code, automaticamente o código será preenchido no seu dispositivo. Veja a Figura 43.

Figura 43. Código de acesso ao aplicativo.

Dessa forma, você poderá acessar seu aplicativo por meio de seu dispositivo. Não esqueça que ele deverá ter o sistema operacional Android.

Saiba mais

A ferramenta possibilita realizar os testes também por meio de um emulador virtual. Para isso, você precisa apenas selecionar a opção *Conectar* e *Emulador*. Acesse o link a seguir e obtenha mais informações.

https://goo.gl/4RdYwe

Referências

KALBACH, J. *Design de navegação Web*: otimizando a experiência do usuário. Porto Alegre: Bookman, 2009.

MIT APP Inventor. *What's New About App Inventor 2?* Massachusetts Institute of Technology, 2017. Disponível em: <http://appinventor.mit.edu/explore/ai2/whats-new.html>. Acesso em: 29 nov. 2018.

Leituras recomendadas

GERBELLI, N. F.; GERBELLI, V. H. P. *App Inventor: s*eus primeiros aplicativos Android. Casa do Código, 2017.

SANTOS, S. S. *Introdução ao App Inventor*. Departamento de Sistemas de Informação. PUC Minas, 2014.

DICA DO PROFESSOR

A ferramenta de desenvolvimento de aplicativos para sistema operacional Android, App Inventor 2, disponibiliza ao desenvolvedor dois tipos de ambiente de programação. Um deles é o Designer, o qual tem todos os componentes, assim como as suas respectivas propriedades para que o *layout* do aplicativo possa ser construído. Outra possibilidade é o Ambiente de Blocos, no qual constam blocos de instruções a serem utilizados para compor a lógica do aplicativo.

Acompanhe o vídeo a seguir e explore cada um desses ambientes, compreendendo as suas funcionalidades.

Conteúdo interativo disponível na plataforma de ensino!

EXERCÍCIOS

1) **O App Inventor tem dois ambientes de desenvolvimento, o de Designer e o de Blocos.**

Assinale a alternativa que traz os componentes pertencentes a cada ambiente citado, respectivamente.

A) SensorAcelerômetro e Controle.

B) CódigoDeBarras e EscolherImagem.

C) Cores e CaixaDeTexto.

D) Procedimentos e Matemática.

E) Legenda e EscolheLista.

2) **O ambiente de desenvolvimento *Designer* tem algumas áreas, nas quais cada uma tem uma função para que o desenvolvedor possa organizar o *layout* de seu aplicativo.**

Assinale a alternativa que traz o nome dessas 5 áreas.

A) Interface de usuário, organização, mídia, mapas e propriedades.

B) Paleta, visualizador, componentes, propriedades e mídia.

C) Conectividade, armazenamento, propriedades, componentes e paleta.

D) Experimental, conectividade, temporizador, blocos, e propriedades.

E) Desenho e animação, mídia, social, visualizador e componentes.

3) **Quais são os componentes que ao serem inseridos possibilitam que estes sejam mostrados no *layout* da esquerda para a direita e do alto para baixo?**

A) Notificador e Botão.

B) Deslizador e OrganizaçãoVertical.

C) OrganizaçãoHorizontal e OrganizaçãoVertical.

D) OrganizaçãoEmTabela e OrganizaçãoHorizontal.

E) SensorDeOrientação e SensorDeProximidade.

4) **A Paleta é organizada por seções, na qual cada seção traz uma gama de componentes relacionados a uma área mais ampla.**

Assinale a alternativa que traz a associação correta entre a seção e o componente.

A) Interface do usuário – Pintura.

B) Sensores – Rectangle.

C) Armazenamento – IniciadorDeAtividades.

D) Desenho e Animação – Bola.

E) Social – *Web*.

5) **O ambiente de blocos tem as instruções para que sejam fornecidas funcionalidade aos componentes que serão inseridos no *layout* do aplicativo.**

Assinale a alternativa que traz um conjunto de blocos desse ambiente de desenvolvimento.

A) Matemática.

B) CloudDB.

C) Ligação.

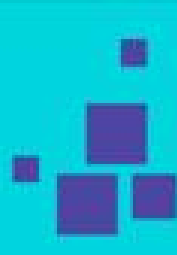
D) Gravador.

E) OrganizaçãoVertical.

NA PRÁTICA

O conhecimento sobre o desenvolvimento de aplicativos se dá não só pelas teorias conceituais, mas pela prática. Por isso, é sempre importante dedicar tempo para alinhar as atividades cotidianas ao uso das ferramentas que estão sendo aprendidas.

Veja a seguir os diversos recursos que o App Inventor disponibiliza, por meio de um aplicativo de tradução desenvolvido por João para uma de suas viagens.

UTILIZANDO O APP INVENTOR A SEU FAVOR

João irá viajar e, como está tendo dificuldade com alguns idiomas, decidiu criar um aplicativo para realizar algumas traduções.

Ele decidiu que o aplicativo terá os seguintes componentes:

Para os componentes do tipo "OrganizadorHorizontal", ele definiu as seguintes configurações de propriedade:

As alterações referentes aos componentes do tipo organizador que contém os componentes "Legenda" e "CaixaDeTexto" variam apenas nas dimensões.

Para os componentes do tipo "Legenda", ele definiu as seguintes configurações de propriedade:

Mudando apenas o título de cada "Legenda".

Para os componentes do tipo "CaixaDeTexTo, ele definiu as seguintes configurações de propriedade:

Em um determinado momento, João teve dúvidas de como o componente "TradutorYandex" funcionava, para isso, ele selecionou a opção "?" que está presente em todos os componentes, e conseguiu visualizar mais informações sobre suas dúvidas.

João descobriu também quais linguagens o componente suporta, por meio do *site*: http://api.yandex.com/translate

Em seu ambiente de blocos, ele inseriu as seguintes instruções:

Após as configurações de layout e das instruções por meio dos blocos, João começou a testar o seu aplicativo, o qual traduzirá as palavras do português para o inglês.

SAIBA MAIS

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Lógica de programação

O segredo do desenvolvimento de *software* está inserido na lógica de programação aplicada na resolução do problema. Veja a seguir o ensinamento da lógica de programação utilizando o App Inventor.

Conteúdo interativo disponível na plataforma de ensino!

Fazendo o primeiro aplicativo

O App Inventor permite a criação dos mais variados tipos de aplicativos, desde que tenha como foco o sistema operacional Android. Veja a seguir as etapas necessárias para a criação de aplicativos utilizando o App Inventor.

Conteúdo interativo disponível na plataforma de ensino!

***Design* com App Inventor**

A complexidade inserida nos aplicativos atualmente demanda que as ferramentas de desenvolvimento tenham recursos cada vez mais robustos. Para isso, veja a seguir, a possibilidade de trazer melhorias para os aplicativos criados no App Inventor.

Conteúdo interativo disponível na plataforma de ensino!